Инженерный вестник Дона, №9 (2021)
ivdon.ru/ru/magazine/archive/n9y2021/7192

Генерация траекторий оптимальных кривых беспилотного летательного аппарата для обхода статического препятствия

Д.Л. Винокурский, К.Ю. Ганьшин, О.С. Мезенцева, Ф.В. Самойлов.

Северо-Кавказский федеральный университет, Ставрополь

Аннотация: В данной работе представлен алгоритм обнаружения и обхода препятствия одиночным беспилотным летательным аппаратом. Предложенный способ представляет собой расширенную версию предыдущей работы с добавлением возможности по предотвращению столкновений на основе построения дополнительной кривой обхода на основе алгоритма де Кастельжо. В этой работе авторы предложили способ генерации траекторий, в котором используются кривые Безье со свойствами годографа Пифагора.
Ключевые слова: беспилотный летательный аппарат, полиномы Бернштейна-Безье, годограф Пифагора, генерация траекторий, обход препятствия, программное движение.

Введение

Траекторное управление - сложная задача, которая зачастую требует описания определённого решения при разработке робототехнических систем. В данной задаче наиболее часто выделяются две подзадачи: генерация оптимальных траекторий и определение методов обхода препятствий. На сегодняшний день в различных публикациях описаны методы построения оптимальных траекторий в зависимости от кинематических и динамических ограничений, влияния окружающей среды и выполняемой задачи.

Основными алгоритмами построения траектории беспилотного летательного аппарата (БПЛА) являются: диаграмма Вороного [1-2], алгоритм Дейкстры [3], метод теории графов [4], методы оптимизации (линейное и нелинейное программирование) [5-6], целочисленное программирование [7-8], методы нечеткой логики и оптимизации траектории генетическими алгоритмами [9-11].

В данной статье авторы используют алгоритм генерации траектории на основе кривых Безье пятого порядка со свойствами годографа Пифагора с разложением по полиномам Бернштейна [12-13]. Применение подобного генератора траекторий обусловлено сниженной вычислительной сложностью задания кривых, легкостью их дифференцирования и возможностью точного определения длины пути. Для успешной работы необходимо иметь возможность перестраивать траектории за как можно меньшее время для избежания столкновений. В данной работе представлен метод определения оптимальной кривой Безье, основанный на предыдущей работе авторов [14], а также модифицированный метод обхода препятствий на основе алгоритма де Кастельжо, предложенный в работе [15]. Описываемый в настоящей работе способ позволяет избегать неподвижных препятствий.

Описанный метод имеет ограничение в виде отсутствия способности обхода динамических препятствий. Траектория кривой представлена как вектор разделения препятствия и транспортного средства в зависимости от времени. Для предотвращения столкновений строится обходная кривая на основе годографа Пифагора.

Генерация траектории

Кривые Безье со свойствами годографа Пифагора выбраны в качестве траекторий следования БПЛА. Если принять во внимание тот факт, что данные кривые третьего порядка (кубики) по сути своей являются лишь одной кривой – кривой Чирнхауза, а значит не обладают должной «гибкостью», было решено использовать кривые пятого порядка (квинтики).

Задание кривых с годографом Пифагора осуществляется за счёт указания первых двух и последних двух контрольных точек (p_0, p_1, p_4, p_5), определяющих кривую Безье пятого порядка. Однако, в случае использования БПЛА, возможно представить задание не четырёх точек, а двух точек (начальную и конечную) и скоростей при них $(p_0, p_5, \dot{p}_0, \dot{p}_5)$; из последних легко вычислить координаты точек p_1 и p_4. Далее производится определение оставшихся точек p_2 и p_3, для чего применятся интерполяция Эрмита. Учитывая, что годографы заданы, как:

$$x'(t) = u^2(t) - v^2(t), \quad y'(t) = 2u(t)v(t) \tag{1}$$

, где $u(t)$ и $v(t)$ можно представить в виде многочленов:

$$\mathrm{u(t) = u_0(1-t)^2 + u_1 2(1-t)t + u_2 t^2} \tag{2}$$

$$v(t) = v_0(1-t)^2 + v_1 2(1-t)t + v_2 t^2 \tag{3}$$

, а контрольные точки p_2 и p_3 определяются, как:

$$p_2 = p_1 + \frac{1}{5}(u_0u_1 - v_0v_1, u_0v_1 + u_1v_0) \tag{4}$$

$$p_3 = p_2 + \frac{2}{15}(u_1^2 - v_1^2, 2u_1v_1) + \frac{1}{15}(u_0u_2 - v_0v_2, u_0v_2 + u_2v_0) \tag{5}$$

Интерполяция Эрмита позволяет определить коэффициенты $u_0, u_1, u_2, v_0, v_1, v_2$:

$$(u_0, v_0) = \pm\sqrt{\frac{5}{2}}\left(\sqrt{|p_1 - p_0| + (p_{1x} - p_{0x})},\ \text{sign}(p_{1y} - p_{0y})\sqrt{|p_1 - p_0| - (p_{1x} - p_{0x})}\right) \tag{6}$$

$$(u_2, v_2) = \pm\sqrt{\frac{5}{2}}\left(\sqrt{|p_5 - p_4| + (p_{5x} - p_{4x})},\ \text{sign}(p_{5y} - p_{4y})\sqrt{|p_5 - p_4| - (p_{5x} - p_{4x})}\right) \tag{7}$$

$$(u_1, v_1) = -\frac{3}{4}(u_0 + u_2,\ v_0 + v_2) + \sqrt{\frac{1}{2}}\left(\sqrt{c + a},\ \text{sign}(b)\sqrt{c - a}\right), \tag{8}$$

, где $c = \sqrt{a^2 + b^2}$,

$$a = \frac{9}{16}(u_0^2 - v_0^2 + u_2^2 - v_2^2) + \frac{5}{8}(u_0u_2 - v_0v_2) + \frac{15}{2}(p_{4x} - p_{1x}),$$

$$b = \frac{9}{8}(u_0v_0 + u_2v_2) + \frac{5}{8}(u_0v_2 + u_2v_0) + \frac{15}{2}(p_{4y} - p_{1y}),$$

$$sign(x) = \begin{cases} 1\text{, при } x > 0 \\ \pm 1\text{, при } x = 0. \\ -1\text{, при } x < 0 \end{cases}$$

Отмечается, что решений всего 4, среди которых следует выбрать наиболее оптимальное. В данном случае выбор оптимального решения производится по критерию наименьшей длины получаемой кривой и наименьшей кривизны в конечных точках. Такой подход выбран в связи с тем, что 4 разных решения интерполяции объединяются в две пары с равными длинами кривых, потому определение лишь наименьшего её значения оказывается недостаточным для однозначного выбора «хорошей» кривой. Длина кривой определяется следующим образом:

$$S = \frac{\sum_{i=0}^{n-1} \sigma_i}{n} \tag{9}$$

, здесь $n=5$, σ_i – коэффициенты параметрической скорости $\sigma(t)$, которые в случае кривой пятого порядка представлены следующим образом:

$$\begin{aligned} \sigma_0 &= u_0^2 + v_0^2 \\ \sigma_1 &= u_0 u_1 + v_0 v_1 \\ \sigma_2 &= \frac{2}{3}(u_1^2 + v_1^2) + \frac{1}{3}(u_0 u_2 + v_0 v_2) \\ \sigma_3 &= u_1 u_2 + v_1 v_2 \\ \sigma_4 &= u_2^2 + v_2^2 \end{aligned} \tag{10}$$

Кривизна плоской кривой задаётся параметрически:

$$k(t) = \frac{[r'(t) \times r''(t)] * z}{|r'(t)|^3} \tag{11}$$

Кривизна в конечных точках:

$$k_{begin} = k(0) = 4\frac{u_0 v_1 - u_1 v_0}{(u_0^2 + v_0^2)^2}, \qquad k_{end} = k(1) = 4\frac{u_1 v_2 - u_2 v_1}{(u_2^2 + v_2^2)^2} \tag{12}$$

Для поиска оптимальной кривой произведём суммирование кривизны конечных точек:

$$k_{sum} = |k_{begin}| + |k_{end}| \tag{13}$$

Далее, определим критерий оптимальной кривой следующим образом:

$$z = \begin{bmatrix} (1,1) \\ (1,-1) \\ (-1,1) \\ (-1,-1) \end{bmatrix}, \quad z_{opt} = \quad z[\underset{i=0,\cdots,3}{\operatorname{argmin}} \; (S_i * k_{sum_i})] \tag{14}$$

Таким образом, определяется сочетание знаков при вычислении коэффициентов $(u_0, v_0), (u_2, v_2)$ такое, что определяется произведение $S * k_{sum}$ для каждого решения интерполяции Эрмита, среди которых находится лишь наименьшее.

Сами же кривые рассчитываются за счёт представления их в форме многочленов Бернштейна [14]:

$$r(t) = \sum_{k=0}^{5} p_k C_5^k (1-k)^{5-k} t^k, \quad t \in [0,1] \tag{15}$$

Для возможности указания реальных временных границ генерации параметрической кривой, используется минимакс-нормализация:

$$t = \frac{T - \min(T)}{\max(T) - \min(T)} \tag{16}$$

, где T – набор дискретных значений времени.

Обход статичного препятствия

Предположим, что препятствие – статичная точка на плоскости $p_o = [x, y], \; p_o \in \mathbb{R}^2, \; x, y = const.$ Определим условие наличия столкновения следующим образом:

$$|r(t) - p_o| \le d \tag{17}$$

где d – скаляр, определяющий радиус безопасности вокруг препятствия.

Зададим новую параметрическую кривую Безье $r_c(t)$, по которой возможно определять наличие пересечения между препятствием и заданной траекторией r(t):

$$r_c(t) = r(t) - p_o = \sum_{k=0}^{5} p_k C_5^k (1-k)^{5-k} t^k - p_o \tag{18}$$

Таким образом, условие наличия столкновения можно переопределить как:

$$\min_{t\in[0,1]} |r_c(t)| \leq d \tag{19}$$

Отсюда, определим минимальную дистанцию между траекторией и препятствием d_{min}, и момент времени данного сближения t_{min} следующим образом:

$$d_{min} = \min_{t\in[0,1]} |r_c(t)|, \quad t_{min} = \operatorname*{argmin}_{t\in[0,1]} |r_c(t)| \tag{20}$$

В виду того, что в процессе полёта затруднительно пересчитывать всю кривую траектории, применим алгоритм де Кастельжо для разделения существующей траектории в двух точках, как описано в [15]:

$$r_{sub}(\tau) = r_c\left(t_{sub}^l + \tau\left(t_{sub}^u - t_{sub}^l\right)\right), \ \tau \in [0,1] \tag{21}$$

Вырезаемый сегмент кривой определяется на интервале $[t_{sub}^l, t_{sub}^u]$, таким образом алгоритм де Кастельжо выполняется на точках $t = t_{sub}^l$ и $t = t_{sub}^u$. Данные параметры возможно определить следующим образом:

$$\begin{bmatrix} t_{sub}^l \\ t_{sub}^u \end{bmatrix} = \begin{cases} \left[t_c, \ \frac{t_{cc}-t_c+\tau_{ds}^l t_c}{\tau_{ds}^l}\right]^T, \text{при } \frac{t_{cc}-t_c}{1-t_c} < \tau_{ds}^l, \\ \left[\frac{t_{cc}-\tau_{ds}^u}{1-\tau_{ds}^u}, \ 1\right]^T, \text{при } \frac{t_{cc}-t_c}{1-t_c} > \tau_{ds}^u, \\ [t_{cc}, \ 1]^T, \text{в остальных случаях} \end{cases} \tag{22}$$

Здесь t_c – момент времени обнаружения препятствия, t_{cc} – момент времени с предполагаемым столкновением. τ – время, отсчитываемое с начала обнаружения препятствия, определяет так же точки разреза основной кривой. τ_{ds}^{l} и τ_{ds}^{u} – границы окна столкновения. Очевидно, что t возможно спроецировать в τ, причём сделать это можно, применив выражение (16):

$$proj(t) = \frac{t - t_{sub}^{l}}{t_{sub}^{u} - t_{sub}^{l}} \tag{23}$$

Определим траекторию обхода следующим образом:

$$r_{det}\left(\tau_{cc}, proj(t)\right) = Ks\left(\tau_{cc}, proj(t)\right)\frac{r_{sub}(\tau_{cc})}{|r_{sub}(\tau_{cc})|} \tag{24}$$

$$s(\tau_{cc}, \tau) = \sum_{k=0}^{n} \bar{s}_k(\tau_{cc}) b_k^n(\tau), \quad \bar{s}_k(\tau) = \begin{cases} \frac{b_k^n(\tau)}{\sum_{j=3}^{n-3}\left(b_k^n(\tau)\right)^2}, & 3 \le k \le n-3 \\ 0 \end{cases} \tag{25}$$

, здесь K– масштабный коэффициент, который определяется условием

$$\min_{\tau \in [\tau_{cc}^{l}, \tau_{cc}^{u}]} \left(\left| r_{\det(\tau_{cc}, \tau)} + r_{sub}(\tau) \right| \right) > d \tag{26}$$

Таким образом, итоговая кривая вместе с кривой обхода, при условии, что t находится в границах времени разделения изначальной траектории $[t_{sub}^{l}, t_{sub}^{u}]$, примет вид:

$$r_{final}(t) = r(t) + r_{det}\left(\tau_{cc}, proj(t)\right) \tag{27}$$

Аппаратно-программный комплекс

Предлагаемый подход экспериментально апробирован на миниатюрном четырёхроторном БПЛА Crazyflie 2.1. В виду малой массы (34 г) и ограниченной вычислительной мощности данного устройства, все вычисления и контроль производится на станции наземного контроля (СНК), представленной в виде персонального комьютера (ПК) и системы радиосвязи.

Обмен данными между СНК и БПЛА производится посредством системы радиосвязи диапазона ISM 2.4 ГГц CrazyRadio PA с заданной пропускной способностью в 2 Мб/с. БПЛА передаёт на СНК сведения о собственном местоположении в локальной навигационной системе (ЛНС) и о собственных углах поворота относительно горизонтальной плоскости земли. Для снижения ошибки позиционирования в ЛНС на БПЛА установлена дополнительно плата расширения с простой системой визуальной одометрии Flow Deck, определяющей смещение БПЛА относительно горизонтальной плоскости земли. ЛНС представляет из себя набор из 8 широкополосных СВЧ-радиомаяков, чьё местоположение относительно друг друга известно, расположенных в вершинах параллелепипеда, который и определяет допустимую область полёта БПЛА. Квадрокоптер использует оценочную функцию на основе расширенного фильтра Калмана, благодаря которому объединяет сведения от бортового приёмника ЛНС, системы визуальной одометрии и собственной инерциальной системы, что позволяет ему достаточно точно определять собственное местоположение в пространстве.

ПК собран на базе процессора Intel 2 поколения i5-2540m, оборудован 8 ГБ DDR3 RAM. Используется операционная система Ubuntu 20.04 LTS, метаоперационная система ROS Noetic, интерпретатор Python 3.6. Программное обеспечение написано на языке программирования Python 3.

Эксперимент

Для проведения испытаний производится предварительная генерация траектории, состоящей из двух кривых Безье с годографом Пифагора. Забор данных о позиции с БПЛА производился каждые 25 мс с дальнейшей фильтрацией в дискретном ФНЧ для получения наиболее гладкой траектории при построении графиков. Препятствие представляет собой виртуальную точку, располагающуюся по координатам *(9.3, 1.9)*, с безопасным радиусом, равным 1 метру. Как видно из рис. 1 и 2, первый разрыв траектории произведён в точке *(9.59, 3.19)*, второй разрыв - в точке *(9.84, 0.97)*, между которыми добавлена новая траектория. В виду того, что кривые Безье с годографом Пифагора после интерполяции Эрмита определяются выражениями (11) - (14), а также невозможно построить в интервале разрыва путь обхода такой, чтобы итоговая кривая в местах разрыва оказалась дифференцируемой, произведено дополнительно переопределение основных секций (начальной и конечной) траекторий.

На рисунке 3 представлены графики скоростей: желаемой (сгенерированная кривая, сиреневая линия) и действительной (скорость реального БПЛА, красная линия). На графиках a, b рис.3 можно увидеть всплески в сгенерированной траектории в окрестностях точек *t=0.45, t=0.65*, а также на графиках c, d в окрестностях точки *t=0.5*, свидетельствующие о наличии “склейки” кривых Безье, однако это не оказывает критического влияния на движение квадрокоптера в виду его инерциальности.

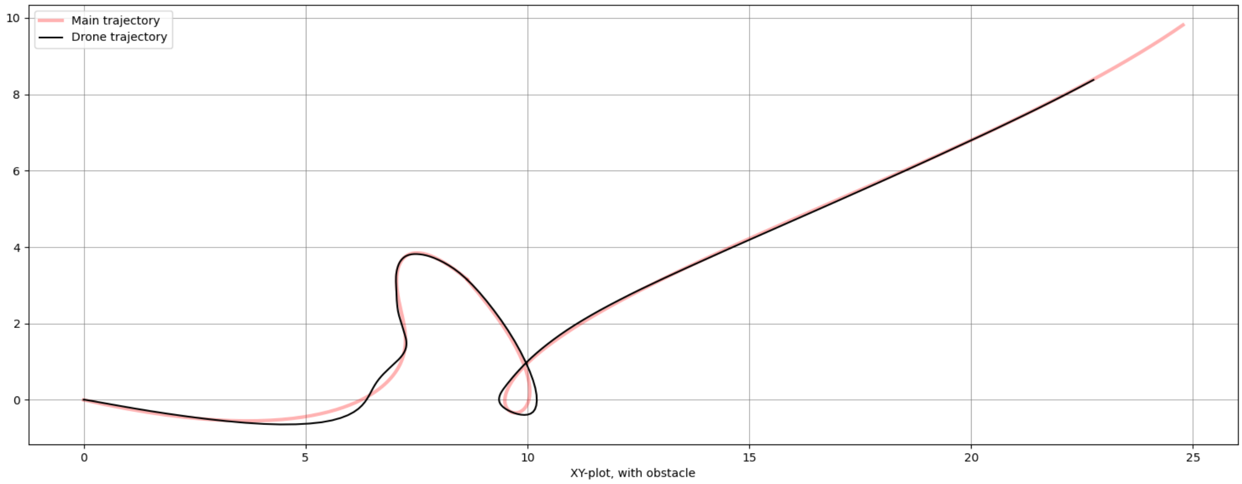

Рис. 1 — Начальная траектория, красным обозначена базовая траектория, чёрным - реальная траектория БПЛА

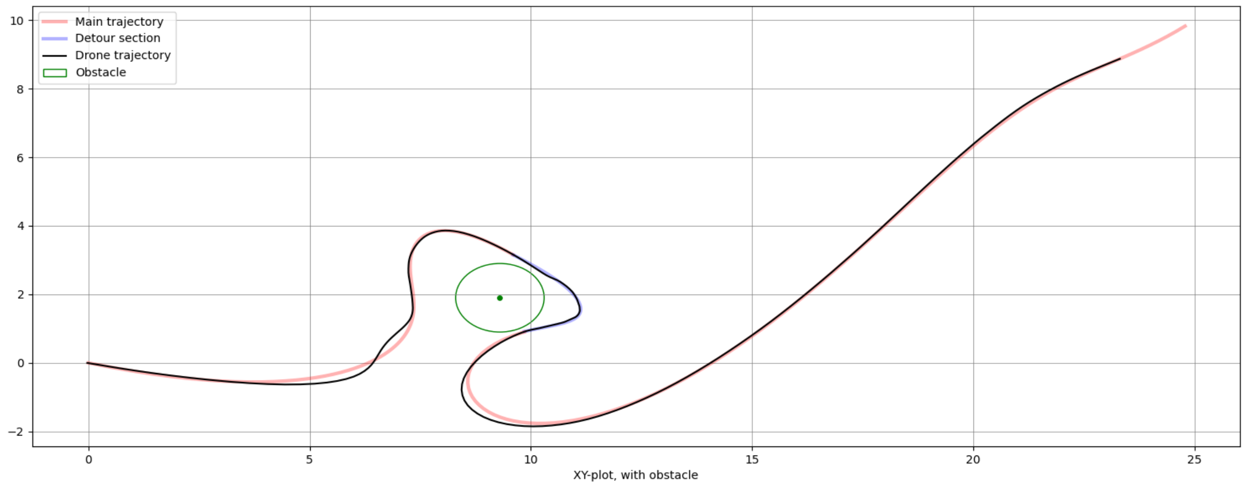

Рис. 2 — Обход препятствия, красным обозначена базовая траектория, синим - обход, зелёным - границы препятствия, чёрным - реальная траектория БПЛА

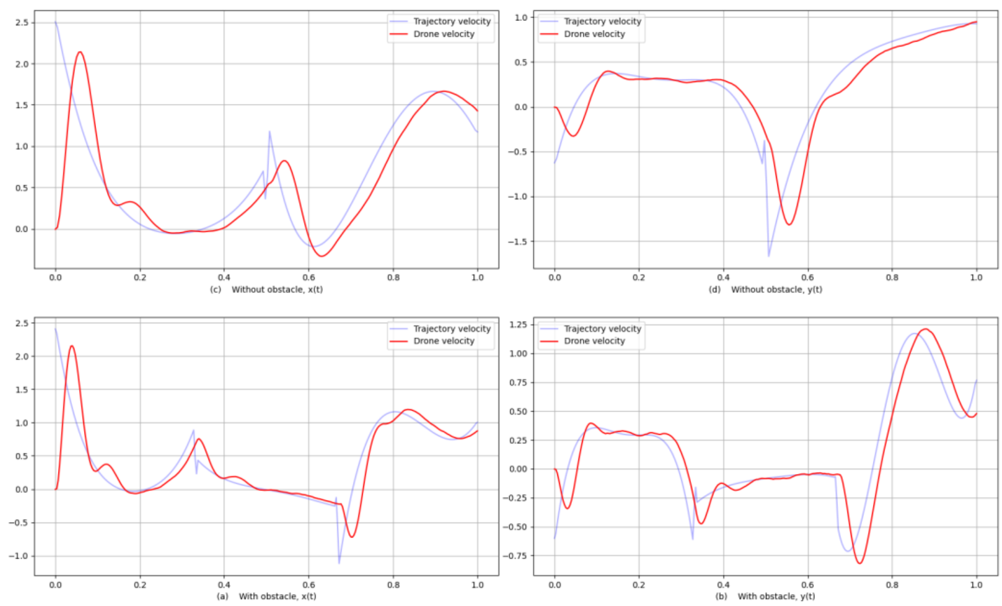

Рис. 3 — a, b - скорости по осям X и Y с препятствием; c, d - скорости по осям X и Y без обнаружения препятствия соответственно

Выводы

В статье был представлен метод поиска оптимальной кривой Безье с годографом Пифагора, который был применён в также представленном алгоритме обнаружения и обхода препятствий на основе алгоритма де Кастельжо. Как показывают результаты представленного эксперимента, выбор оптимальной кривой Безье работает удовлетворительно для задач построения траекторий полёта БПЛА: перегибы функций возникают не близ точек начала и конца кривых Безье и обладают малой кривизной, которая не вызывает больших расхождений между желаемыми и действительными координатами БПЛА. Тем не менее, возможно дальнейшее развитие метода оптимизации с возможностью указания требуемой кривизны для соответствия полётным характеристикам модели БПЛА.

Предложенный способ обхода препятствий позволяет изменять уже существующую траекторию для предотвращения столкновений. Как показывают результаты испытаний, в местах разрывов и соединений кривых Безье наличествуют выбросы функций уже при взятии первой производной. Подобное явление говорит о необходимости внедрения методов оптимизации соединения кривых по производным второго порядка, что приведёт к меньшим энергетическим затратам БПЛА. В будущем планируется расширение алгоритма и его улучшение с добавлением возможности обхода динамических препятствий и генерации более плавной траектории.

Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 20-31-90060.

Литература

1. Tong H. et al. Path planning of UAV based on Voronoi diagram and DPSO // Procedia Engineering. – 2012. – Т. 29. – pp. 4198-4203.

2. Chen X., Chen X. The UAV dynamic path planning algorithm research based on Voronoi diagram // The 26th chinese control and decision conference (2014 ccdc). – IEEE, 2014. – pp. 1069-1071.

3. Medeiros F. L. L., Da Silva J. D. S. A Dijkstra algorithm for fixed-wing UAV motion planning based on terrain elevation // Brazilian Symposium on Artificial Intelligence. – Springer, Berlin, Heidelberg, 2010. – pp. 213-222.

4. Toan T. Q., Sorokin A. A., Trang V. T. H. Using modification of visibility-graph in solving the problem of finding shortest path for robot //2017 International Siberian Conference on Control and Communications (SIBCON). – IEEE, 2017. – pp. 1-6.

5. Ragi S., Mittelmann H. D. Mixed-integer nonlinear programming formulation of a UAV path optimization problem //2017 American Control Conference (ACC). – IEEE, 2017. – pp. 406-411.

6. Borrelli F. et al. MILP and NLP techniques for centralized trajectory planning of multiple unmanned air vehicles //2006 American Control Conference. – IEEE, 2006. – p. 6.

7. Glover F. Improved linear integer programming formulations of nonlinear integer problems //Management Science. – 1975. – Т. 22. – №. 4. – pp. 455-460.

8. Ioan D. et al. Mixed-integer programming in motion planning //Annual Reviews in Control. – 2020. pp.85-87.

9. Silva Arantes J. et al. Heuristic and genetic algorithm approaches for UAV path planning under critical situation // International Journal on Artificial Intelligence Tools. – 2017. – Т. 26. – №. 01. – pp. 1760008.

10. Chen G., Cruz J. B. Genetic algorithm for task allocation in UAV cooperative control //AIAA Guidance, Navigation, and Control Conference and Exhibit. – 2003. – p. 5582.

11. Wilburn B. K., Perhinschi M. G., Wilburn J. N. A modified genetic algorithm for UAV trajectory tracking control laws optimization //International Journal of Intelligent Unmanned Systems. – 2014. pp. 58-90.

12. Vanegas G. et al. Smooth 3D path planning for non-holonomic UAVs //2018 7th International Conference on Systems and Control (ICSC). – IEEE, 2018. – pp. 1-6.

13. Bestaoui Y. 3D flyable curves for an autonomous aircraft //AIP Conference Proceedings. – American Institute of Physics, 2012. – Т. 1493. – №. 1. – pp. 132-139.

14. Винокурский Д. Л. и др. Планирование траектории группы беспилотных летательных аппаратов с использованием годографа Пифагора и составных кривых Бернштейна-Безье на плоскости //Вестник КРАУНЦ. Физико-математические науки. – 2020. – Т. 31. – №. 2. – С. 70-78.

15. Mehdi S. B. et al. Collision avoidance through path replanning using Bézier curves //AIAA Guidance, navigation, and control conference. – 2015. – p. 0598.

References

1. Tong H. Procedia Engineering. 2012. T. 29. pp. 4198-4203.

2. Chen X., Chen X. The 26th chinese control and decision conference (2014 ccdc). IEEE, 2014. pp. 1069-1071.

3. Medeiros F. L. L., Da Silva J. D. S. Springer, Berlin, Heidelberg, 2010. pp. 213-222.

4. Toan T. Q., Sorokin A. A., Trang V. T. H. 2017 International Siberian Conference on Control and Communications (SIBCON). IEEE, 2017. pp. 1-6.

5. Ragi S., Mittelmann H. D. 2017 American Control Conference (ACC). IEEE, 2017. pp. 406-411.

6. Borrelli F. et al. 2006 American Control Conference. IEEE, 2006. p. 6.

7. Glover F. Management Science. 1975. T. 22. №. 4. pp. 455-460.

8. Ioan D. et al.Annual Reviews in Control. 2020. pp.85-87.

9. Silva Arantes J. International Journal on Artificial Intelligence Tools. 2017. T. 26. №. 01. p. 1760008.

10. Chen G., Cruz J. B. AIAA Guidance, Navigation, and Control Conference and Exhibit. 2003. p. 5582.

11. Vanegas G. et al. Smooth 3D path planning for non-holonomic UAVs 2018 7th International Conference on Systems and Control (ICSC). IEEE, 2018. pp. 1-6.

12. Vanegas G. 2018 7th International Conference on Systems and Control (ICSC). IEEE, 2018. pp. 1-6.

13. Bestaoui Y. AIP Conference Proceedings. – American Institute of Physics, 2012. T. 1493. №. 1. pp. 132-139.

14. Vinokurskij D. L. Vestnik KRAUNC. Fiziko-matematicheskie nauki. 2020. T. 31. №. 2. pp. 70-78.

15. Mehdi S. B. et al. Guidance, navigation, and control conference. 2015. p. 0598.